情報保護學會論文誌
第14券 第3號, 2004. 6

SSFNet을 이용한 네트워크 보안 시뮬레이터에서 동적 시뮬레이션 방법

윤 주 범, 박 응 기, 임 을 규
국가보안기술연구소

Dynamic Simulation on a Network Security Simulator using SSFNet

Joo Beom Yun, Eung Ki Park, Eul Gyu Im
National Security Research Institute

요 약

최근 사이버 테러에 대한 관심이 고조되면서 네트워크 보안 시뮬레이터가 필요하게 되었다. 네트워크 보안 시뮬레이터는 침입 행위 및 방어 행위를 모델링하여 침입에 대한 피해 정도, 방어 대책의 효과성 등을 파악하기 위한 도구이다. 이 도구를 통해서 사이버 테러에 대한 예방과 복구가 가능하기 때문이다. 이를 위해 기존의 시뮬레이터들은 시뮬레이션 수행 전에 모든 시나리오를 작성해 놓고 시뮬레이션을 수행하였다. 그러나 사람의 판단 및 행위를 모델링하지 못한 시뮬레이션은 정확한 결과를 나타내지 못하였다. 따라서 본 논문에서는 기존 네트워크 보안 시뮬레이터에 동적 시뮬레이션 요소를 첨가함으로써 정확히 네트워크 침입 및 방어 행위를 표현하고자 하였다. 또한 이를 위한 시뮬레이터 구조 변경 방법을 제안하였다. 시뮬레이터를 구현한 후에는 슬래머 웜의 시뮬레이션을 수행하여 기능이 올바로 구현되었음을 확인하였다.

ABSTRACT

Recently, a network defense simulator becomes essential in studying cyber incidents because the cyber terror become more and more interesting. The network defense simulator is a tool to estimate damages and an effectiveness of a defense mechanism by modeling network intrusions and defense mechanisms. Using this tool, users can find efficient ways of preventing a cyber terror and recovering from the damage. Previous simulators start the simulation after entire scenario has made and been loaded to simulation engine. However, in this way it can't model human judgement and behavior, and it can't simulate the real cyber terror very well. In this paper, we have added a dynamic simulation component to our previous network security simulator. This component improved accurate modeling of network intrusions and defense behaviors. We have also proposed new modified architecture of the simulation system. Finally we have verified correct simulation results from slammer worm simulation.

Keywords : *SSFNet, Network security simulator, dynamic simulation*

Ⅰ. 서 론

최근 사이버 테러에 대한 관심이 고조되면서 네트워크 시뮬레이션에 정보보호 요소를 가미한 네트워크 보안 시뮬레이터의 개발이 요구되고 있다. 그러나 네트워크의 방대함, 침입과 방어의 복잡성 및 다양성의 표현 어려움 등으로 인하여 아직까지 많은 연구의 진전이 없는 분야이기도 하다. 특정 침입에 대한 네트

접수일 : 2004년 3월 19일 ; 채택일 : 2004년 5월 11일
† 주저자, {netair,ekpark,imeg}@etri.re.kr

워크 행동 변화 및 방어 기법에 대한 자료를 얻기 위하여, 실제 네트워크에 사이버 침입을 시도하고 그때의 네트워크 행동 변화를 관찰하는 것이 가장 좋은 방법이다. 그러나 발생 가능한 모든 침입을 시도하여 필요한 자료를 얻는 것은 위험성이 존재할 뿐만 아니라 현실적으로 여러 가지 제약사항이 따른다. 이에 대한 대안으로 시뮬레이션 기법을 들 수 있으며, 이를 수행하는 네트워크 보안 시뮬레이터의 개발 노력은 계속되어 왔다. 첫째로, 개념이나 이론을 테스트하기 위하여 시뮬레이터를 제작·사용하였다. 예를 들어, Smith와 Bhattac Harya[1]는 소규모 네트워크에서 침입차단시스템 위치에 따른 성능을 평가하기 위해 시뮬레이터를 사용하였다. 또한 시뮬레이터는 네트워크 형태(topology)와 성능을 평가하기 위해 사용되었다[2,3]. 그러나, 대규모 사이버 테러에 관한 시뮬레이션은 Mostow, et al.[4]의 IAS(Internet Attack Simulator)에서 시작되었다. IAS에서는 3가지 침입 시나리오(서비스 거부 공격, 허가되지 않은 접근, 속임(spoofing))를 시뮬레이션할 수 있는 시뮬레이터를 제작하였다. Donald Welch, et al.[5]의 정보전 시뮬레이션 프레임워크에서는 암호 가로채기(password sniffing), 침입차단시스템 효과에 관한 시뮬레이션을 수행하였다. 그러나 이와 같은 네트워크 보안 시뮬레이터들이 현재 실세계 네트워크와 사이버 침입 및 방어를 제대로 표현하는 수준은 아니라고 판단된다. SSFNet[8](Scalable Simulation Framework Network Models)을 이용하여 가상 침입을 모델링하고 이에 따른 네트워크 행동 변화 시뮬레이션을 수행하는 시뮬레이터는 [6]에서 제안되었다. SSFNet은 실제 인터넷을 모델링하고 시뮬레이션하기 위해서 프로토콜들(IP, TCP, UDP, BGP4, OSFP 등)과 네트워크 구성 요소(호스트, 라우터, 링크, 랜 등)를 클래스로 구현한 자바 모델이다. 하지만 [6]의 시뮬레이터는 사전에 모든 것이 결정되어 정해진 상황에 대해서만 시뮬레이션을 수행하고 사람의 행위와 같은 동적 요소는 표현할 수 없었다. 그러나 사이버 침입 및 방어 시뮬레이션은 분명 사람의 행위가 가미되는 동적인 시뮬레이션이다. 이에 대한 방안을 본 논문에서 제시하겠다.

본 논문은 다음과 같이 구성된다. 2장에서는 기존 시뮬레이터의 구조를 제시하고, 3장에서는 동적 시뮬레이션을 구현하기 위해 제안되는 새로운 시뮬레이터 구조와 새로운 시뮬레이터 구현에 요구되는 요소에 대해 설명하고, 4장에서는 시뮬레이션 결과를 제시하겠다. 마지막으로 5장에서 결론을 제시하겠다.

II. 기존 네트워크 보안 시뮬레이터 구조

기존에 제안된 네트워크 보안 시뮬레이터는 그림 1과 같이 클라이언트-서버 모델을 바탕으로 설계되었다. 이 모델을 채택한 이유는 성능 향상을 위한 병렬 및 분산 시뮬레이션이 가능하기 때문이다. 시뮬레이션 클라이언트는 사이버 침입 및 방어 시나리오를 작성하고, 시뮬레이션 결과를 보여주는 그래픽 사용자 인터페이스 프로그램이다. 시뮬레이션 서버는 시뮬레이션 엔진을 가지고 있으며, 시나리오를 DML (Domain Modeling Language) [7] 로 바꾸어 주는 DML 변환기, 시뮬레이션 클라이언트와 통신하기 위한 데몬 등 시뮬레이션 수행에 필요한 핵심 요소들을 가지고 있다.

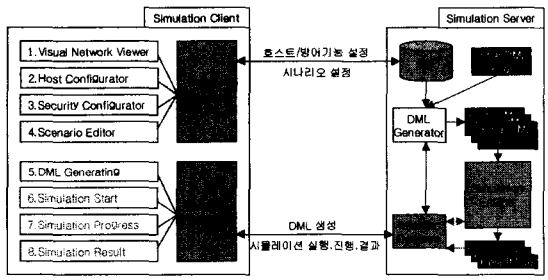


그림 1. 기존 네트워크 보안 시뮬레이터 구조

사용자는 시뮬레이션 클라이언트 프로그램을 실행시킨 후 시뮬레이션 클라이언트 프로그램에서 네트워크를 편집한다. 네트워크 편집이란 네트워크 장비(라우터, 스위치 등)들을 배치하고 호스트 등을 실선으로 연결하여 원하는 형태(topology)로 구성하는 것이다. 네트워크를 구성한 후에는 네트워크 안에 존재하는 호스트의 속성을 설정한다. 그리고 나서 시뮬레이션을 수행하고자 하는 침입 시나리오를 편집한다. 시나리오 생성은 단위 시나리오(actor)들을 조합하여 이루어지는데, 시나리오를 시뮬레이션 엔진이 이해하도록 하기 위해 DML로 변환한다. 지금까지 사용자가 설정한 정보는 데이터베이스 커넥터를 통해 시뮬레이션 서버에 전달된다. 그 후 사용자가 DML 생성을 지시하면 그 요구가 서버로 전달되고, DML 생성기는 네트워크 및 호스트 정보와 시나리오 정보를 합하여 시뮬레이션 엔진이 이해할 수 있는 최종

DML 파일을 생성한다. 그 후 클라이언트의 시뮬레이션 시작 명령에 의해 서버에서는 시뮬레이션이 수행되고 수행된 결과들은 클라이언트로 보내져서 그래픽하게 보여준다.

Ⅲ. 제안하는 네트워크 보안 시뮬레이터 구조

1. 새로운 네트워크 보안 시뮬레이터 구조

동적 시뮬레이션을 위한 새로운 네트워크 보안 시뮬레이터의 구조는 그림 2와 같다. 새로운 구조도 클라이언트-서버 모델을 바탕으로 하였으며 시뮬레이션 클라이언트는 변함이 없다. 하지만 시뮬레이션 서버에 시뮬레이션 엔진과 데몬이 한 개의 모듈로 합쳐졌으며 중간에 생성되는 DML 내용이 기존 구조처럼 많지 않다. 오직 네트워크 연결에 대한 정보만이 DML 구조에 포함된다. 그리고 시뮬레이션에 사용되는 호스트 정보 및 시뮬레이션 시나리오 정보는 별도의 데이터베이스에 저장된다. 이와 같이 DML에 호스트 정보 및 시뮬레이션 시나리오를 제외한 것은 DML은 시뮬레이션 수행 중에 수정이 불가능하기 때문이다. 따라서 수정이 요구되지 않는 네트워크 구조 정보만을 DML에 포함하고 시뮬레이션 중 수정이 필요한 호스트 정보 및 시나리오는 수정이 용이한 별도 데이터베이스에 저장한다. 이것이 기존 시뮬레이터 구조와 가장 큰 차이점이다.

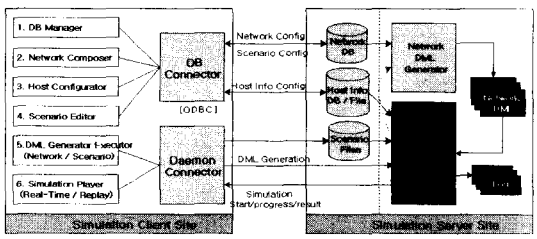


그림 2. 새로운 시뮬레이터 구조

2. 구현을 위한 새로운 기능

제안하는 네트워크 보안 시뮬레이터는 그림 3과 같이 트리 형태의 시나리오 표현을 사용한다. 이 표현은 침입의 최종 목표, 즉 마지막으로 수행되는 노드를 루트 노드로 구성하고, 그 루트 노드에 침입을 수행하기 위해 선행되어야 하는 침입들을 하위 노드로 구성한다. 우리는 이 트리 표현에 실행 순서 개념을 부가하여 실행 순서에 따라 트리를 구성하였다. 먼저 트리의 모든 노드는 계층적 레벨과 실행 순서에 따라 단계번호를 부여한다. 단계번호는 그림 3과 같이 구성된다. 트리의 노드에 단계 번호를 부여하여 표현하면 노드의 트리 구조상의 위치를 확인하기 쉽고, 또한 시뮬레이션 과정에서 노드의 실행순서가 명확해진다.

```
1 시스템 침입 scenario
 1.1 시스템 정보 수집 (SequentialAND)
  1.1.1 Network Scan
  1.1.2 Port Scan
  1.1.3 OS Identification
 1.2 Buffer Overflow 침입(SequentialAND)
  1.2.1 IIS ISAPI 버퍼 오버플로우 침입
  1.2.2 관리자 권한 로그인
  .....
```

그림 3. 트리 형태의 시나리오 표현

트리를 구성하는 노드는 기능에 따라 크게 두 가지로 구분된다. 액터 노드(Actor Node)는 하위노드를 가지지 않는 트리의 종단노드로써, 실제 이벤트를 수행한다. 노드의 이름은 이벤트의 이름으로 표기한다. 연산 노드(Operation Node)는 반드시 하나 이상의 액터 노드나 다른 연산 노드를 하위 노드로 가지며, 하위 노드의 실행순서를 지정하고, AND와 OR 같은 논리연산의 의미를 가진다. 이러한 연산 노드는 시나리오 작성시 액터 노드들 사이의 관계를 보다 잘 표현할 수 있도록 도와준다. 연산 노드에는 순차(Sequential) AND, 순차(Sequential) OR, 병렬(Parallel) AND, 병렬(Parallel) OR가 존재한다.

그림 3과 같은 시나리오를 시뮬레이션할 경우 그림 4와 같이 PortScan 단계 중에 시뮬레이션 정지를 선택하면 PortScan 단계가 끝나고 시뮬레이션이 정지된다. 그리고 나서 시뮬레이션 화면에서 네트워크 설정 정보를 수정한 후 다시 시뮬레이션을 재시작시키면 OS Identification 단계부터 시뮬레이션이 재시작된다.

그림 4에서와 같이 시뮬레이션을 일시 정지 시킨 후 재시작하기 위해서는 시뮬레이션 수행 정보를 저장해 두어야 한다. 시뮬레이션을 수행하면 매 수행시간마다 시뮬레이션 객체인 각 호스트별로 독특한 설정 정보를 가지게 된다. 동적 시뮬레이션을 위해서는 이 호스트별 특정 정보를 시뮬레이션 종료 시까지 유

	Step		1	1.1	1.1.1	1.1.2	1.1.3
	Actor		Scenario	SequantialAND	NetworkScan	PortScan	OSIdentification
	BreakPoint					1	

(1) 사용자의 시뮬레이션 중단 요청
(2) 중단요청(value : 1) 체크된 step의 Actor는 지속적으로 수행
(3) Actor의 수행 완료시 재시작 위치를 확인(value : 2)하고 중단
(4) 재시작시 위치를 확인(value : 0)하고 시작

(3)BreakPoint 값을 2로 변경 시뮬레이션 중단

(4)BreakPoint값을 0으로 변경 시뮬레이션 진행

그림 4. 시나리오 진행 예제

지해야 하는데 이 설정 정보가 저장되어 있는 곳이 각 호스트의 SFS(Simulated File System)이다.

즉, 실제 호스트에 존재하는 파일 시스템과 시뮬레이션 수행에 필요한 정보를 호스트 상태 정보를 각 호스트별로 분류해서 놓은 것을 그 호스트의 SFS라고 한다. 따라서 각 호스트의 SFS에는 현재 호스트 상태, 해당 호스트의 파일 정보 등을 저장하고 있고, 이것을 통해 시뮬레이션 엔진이 개별 호스트의 상태 및 파일 시스템 정보 등을 확인할 수 있는 것이다. 개별 호스트의 SFS 정보를 DB로 만든 것이 Simulated File System 데이터베이스이고 이 데이터베이스에 접근하기 위해 SFS API를 구현하였다. SFS API에는 각 호스트의 파일 생성 및 삭제, 파일 존재유무 검사, 파일 이름 바꾸기, 파일 내용 읽기, 파일 내용 쓰기 등의 함수를 지원한다.

본 논문의 시뮬레이터에서 호스트는 일반적인 네트워크 상에 존재하는 호스트와 동일한 역할을 수행하는 가상의 노드이다. 그러므로 각각의 호스트는 자신의 호스트에서 실행되는 운영체제 정보, 하드웨어 정보, 네트워크 상태, 침입차단시스템 정책 설정 등의 정보를 가지고 있어야 하고 호스트간의 통신이 가능하여 서로의 정보를 교환할 수 있어야 한다. 제안하는 시뮬레이션 시스템의 호스트에는 actor라고 하는 사이버 침입 및 방어 행위를 모델링하는 모듈이 존재한다. 시뮬레이션의 수행을 위해서는 각각의 호스트들을 초기화시키고, 호스트간의 통신을 가능하게 하며 호스트의 상태를 저장하는 모듈이 필요하다.

그림 5에서와 같이 호스트는 SFS, Actor, Service, Executor에 의해 모델링된다. Executor는 각 호스트에 존재하여 실제 호스트의 커널과 유사한 2가지 역할을 한다. 우선 시뮬레이션이 실행되어 네트워크를 초기화할 때 각 호스트의 속성을 SFS로부터 읽어와서 초기화하는 역할을 한다. 그리고 시뮬레이션 엔진에서 Actor를 생성하라는 요청을 받으면 해당 Actor를 동적으로 생성하는 역할과 각 호스트 간에 통신을 하는 역할을 한다.

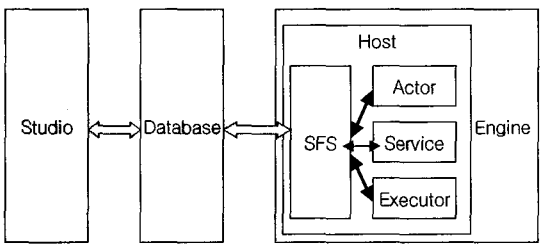


그림 5. 호스트 모델링 개념도

Actor 모듈은 사이버 침입 및 방어 행위를 기술한다. 매번 새로운 시나리오를 시뮬레이터가 이해할 수 있는 언어로 작성한다는 것은 어렵기 때문에 개발자가 미리 정의된 단위 시나리오(Actor모듈)들을 작성해 놓고, 사용자가 이것을 이용하여 시나리오를 작성하도록 한다. 마지막으로 Service는 각 호스트에서 제공되는 서비스(http, ftp, telnet 등)를 실제 프로토콜에 유사하도록 구현한 것이다.

Ⅳ. 시뮬레이션 결과

본 절에서는 기존의 시뮬레이션 결과와 동적인 요소를 첨가한 시뮬레이션 결과를 비교한 내용을 제시한다. 우선 시뮬레이션을 수행한 네트워크 환경은 그림 6과 같다. 그림 6과 같이 시뮬레이션 수행 네트워크는 서브네트워크 14개로 이루어진 1750개의 노드가 모델링된 대규모 네트워크이다. 호스트들에 대한 설정은 1750개 호스트 모두 슬래머 웜의 감염이 가능하도록 MS-SQL 서버가 설치되어 있고, 모두 취약점이 존재하도록 설정하였다. 따라서 시간이 지남에 따라 모든 호스트가 감염이 가능하다. 시뮬레이션은 7번 네트워크의 침입자 컴퓨터에서 1번 네트워

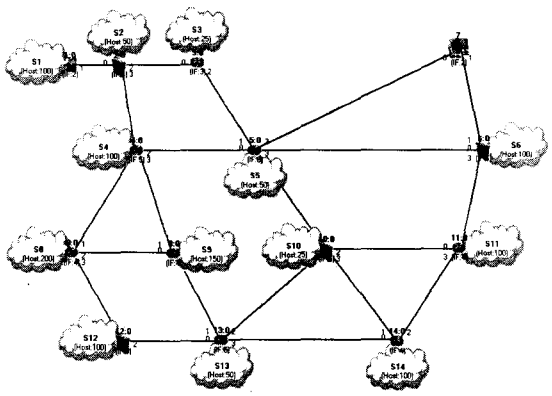


그림 6. 시뮬레이션 네트워크 구조

크(S1)의 취약 시스템에 웜을 유포함으로써 이루어진다. 이때 시뮬레이션 시간 75초 동안 웜에 감염된 호스트 수와 루트 DNS 서버가 받는 초당 패킷 수를 측정하였다.

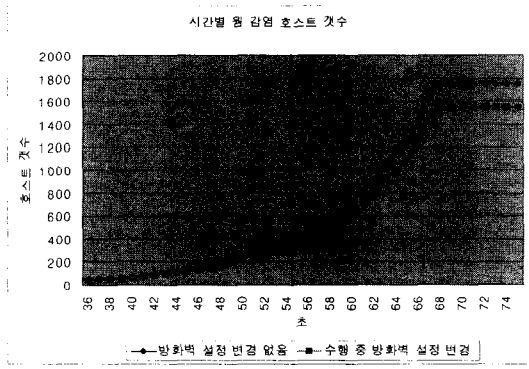


그림 7. 웜 감염 호스트 갯수 그래프

그림 7은 전체 1750개의 호스트 개수 중 시간이 지남에 따라 몇 개의 호스트가 웜에 감염되었는지를 나타내는 그래프이다. 시뮬레이션 수행 중 어떤 사용자 입력도 없는 상태와 수행 중 200개의 호스트를 보호하는 침입차단시스템의 웜 접근 포트를 차단하여 시뮬레이션한 결과이다. 침입차단시스템의 정책 변경은 30초에서 이루어졌으며 그림 7에서 보면 30초 이후 감염 수가 조금씩 차이나는 것을 볼 수 있다. 이것은 침입차단시스템으로 보호된 네트워크에 웜이 유포되지 않기 때문이라고 판단된다. 그림 7을 보면 방화벽 설정을 하지 않은 경우 시뮬레이션을 시작한 후 68초 후에 전체 네트워크가 웜에 감염된 것을 볼 수 있다. 방화벽 설정을 한 경우도 대략 70초쯤에는 보호되는 영역을 제외하고는 전체 네트워크가 웜에 감염된 것으로 분석된다. 그림 7과 그림 8에서 36초 이후부터 결과가 제시된 것은 35초까지의 데이터가 큰 의미가 없기 때문이다.

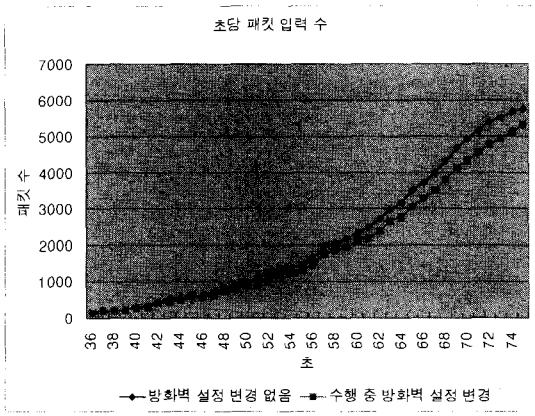


그림 8. 루트 DNS의 초당 패킷 수(packets/sec)

그림 8은 서브네트워크 1에 존재하는 루트 DNS의 초당 패킷 수이다. 이 그림에서도 볼 수 있듯이 침입차단시스템이 적용되어 네트워크가 보호되면 웜에 감염된 호스트의 수가 적어지고 그에 따라 루트 DNS에서 받아들이는 패킷 수도 적어짐을 알 수 있다. 그림 7과 그림 8을 보면 웜이 전체 네트워크를 완전히 감염시킨 70초 이후부터는 DNS의 초당 패킷 입력 수 증가가 완만히 일어나는 것으로 판단된다. 그러나 웜 감염수 그래프와 같이 70초 이후가 평평해지지 않는데 이것은 네트워크의 지연(delay) 및 네트워크 대역폭(bandwidth) 때문으로 생각된다. 그림 7과 그림 8에서 알 수 있듯이 본 논문에서 제안한 동적 요소가 제대로 반영되었음을 알 수 있다.

V. 결론 및 향후과제

사이버 테러를 좀 더 실제 상황에 가깝도록 모델링하고 시뮬레이션하기 위해서는 사람에 의한 동적 시뮬레이션 요소를 반영하여야 한다. 본 논문에서는 시뮬레이터에 호스트 정보 및 시나리오 정보를 저장해 놓고 변경을 허용함으로써 시뮬레이션 수행 중 동적인 변경 사항을 반영하도록 하였다. 또한 슬래머 웜을 모델링한 실험을 통하여 동적 시뮬레이션 기능이 정상적으로 작동함을 확인하였다. 그러나, 아직까지 침입차단시스템 정책 변경 외에는 실험해 보지 못

하였다. 따라서 많은 실험을 통해 동적 변경이 가능한 기능과 그렇지 못한 기능을 판별하고 더 많은 동적 요소가 지원되도록 연구해야 할 것으로 생각된다.

참 고 문 헌

[1] Smith, R and Bhattac Harya, "Firewall Placement In a Large Network Topology" in Proc. 6th IEEE workshop on Future Trends of Distributed Computing Systems, 1997.

[2] Breslau, L., et al., Advances in Network Simulation Computer, Col. 33, No.5, May 2000.

[3] Optimum Network Performance, OPNET Modeler, http://www.opnet.com/products/modeler/home.html, March 2001.

[4] John R. Mostow, John D. Roberts, John Bott, Integration of an Internet Attack Simulator in an HLA Environment, Proc. IEEE workshop on Information Assurance and Security, West Point, NY, June 2000.

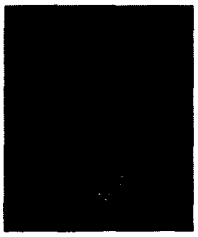
[5] Donald Welch, Greg Conti, Jack Marin, A framework for an Information Warfare Simulation, Proc. IEEE workshop on Information Assurance and Security, June 2001.

[6] 윤주범, 서정택, 이상훈, 이철원, "사이버전 시뮬레이션 프레임워크", 제12회통신정보합동학술대회Proceedings, 2002년 4월.

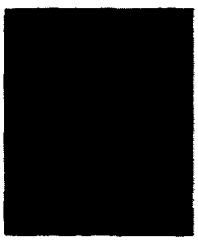
[7] "Domain Modeling Language(DML)", http://www.ssfnet.org/homePage.html.

[8] "Scalable Simulation Framework Network Models", hhttp://www.ssfnet.org

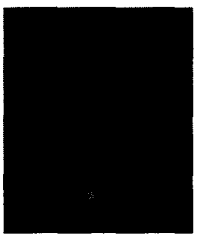
〈著者紹介〉



윤 주 범 (Joo Beom Yun) 정회원
1999년 2월: 고려대학교 컴퓨터학과 졸업
2001년 2월: 서울대학교 컴퓨터공학과 석사
2001년 3월~현재: 국가보안기술연구소 연구원
〈관심분야〉 정보보호, 통신보안



박 응 기 (Eung Ki Park) 정회원
1988년 2월: 중앙대학교 전자계산학과 석사
1988년 2월 ~ 2000년 1월 : 한국전자통신연구원 선임연구원
2000년 1월 ~ 2000년 4월 : 국가보안기술연구소 책임연구원
2000년 4월 ~ 2002년 11월 : (주)니츠 기술이사
2002년 11월~현재: 국가보안기술연구소 책임연구원
〈관심분야〉 정보보증, 컴퓨터 네트워크 보안



임 을 규 (Eul Gyu Im) 정회원
1999년 2월: 고려대학교 컴퓨터학과 졸업
2001년 2월: 서울대학교 컴퓨터공학과 석사
2001년 3월~현재: 국가보안기술연구소 연구원
〈관심분야〉 정보보호, 통신공학